Hardware Acceleration (GPU)

> This section only covers setup for Nvidia GPUs

1. Finding a suitable GPU

The recommended way for hardware acceleration is to use a dedicated GPU. The usual manufacturers are AMD and Nvidia. However, experience has shown, that despite Nvidia's flawed Linux support regarding drivers, it is still the more performant and stable option, for transcoding media files (specifically HEVC). If you are wondering, which GPU is affordable and also supports all the common codecs, I can recommend you to take a look at the following matrix provided by Nvidia: https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new. Make sure, that your GPU supports at least H.265 (4K YUV 4:2:0), since there are still a lot of clients, that don't support the rather new codec. Keep in mind, that using an old "Gaming" GPU might not be the best option for you, because they consume a lot of power and furthermore are locked to a maximum of 3 concurrent sessions. So if you have more than three people streaming video at any given time, you'll run into problems. This leaves Nvidia's Professional/Datacenter GPU's. My recommendation for a cheap, low power consuming and free of restrictions GPU is the Nvidia Quadro P2000. With its 5GB of GDDR5 VRAM, it has enough power, to allow a whole family to stream HQ content. If you know, that you don't exceed the session limit, the cheaper Nvidia Quadro P400 might be of interest to you. With that out of the way, you can continue to install the Nvidia drivers.

2. Installing NVIDIA drivers

Before installing the drivers, make sure, that you meet all requirements:

- ☑ Supported kernel version (https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#system-requirements)
- ☑ CUDA capable GPU
- ☑ NVENC and NVDEC cabable GPU

To start off, make sure, that your `/etc/apt/sources.list` file includes the `contrib` and `non-free` repositories. E.g. the following:

```
deb http://deb.debian.org/debian/ bullseye main contrib non-free
deb-src http://deb.debian.org/debian/ bullseye main contrib non-free
deb http://deb.debian.org/debian/ bullseye-updates main contrib non-free
deb-src http://deb.debian.org/debian/ bullseye-updates main contrib non-free
```

Next, update all packages and install the `nvidia-driver` package.

```
apt update
apt -y install nvidia-driver firmware-misc-nonfree
```

Make sure, that the default nouveau drivers are blacklisted, by running `cat /etc/modprobe.d/nvidia-blacklists-nouveau.conf`

```
# You need to run "update-initramfs -u" after editing this file.


# see #580894
blacklist nouveau
```

If all is set, reboot your machine. You should be able, to verify the loaded drivers now, by running `<span class="kw2">lsmod</span> <span class="sy0">|</span> <span class="kw2">grep</span> nvidia`:

```
nvidia_uvm          1273856  0
nvidia_drm            73728  0
drm_kms_helper       278528  1 nvidia_drm
nvidia_modeset      1146880  1 nvidia_drm
nvidia             40828928  2 nvidia_uvm,nvidia_modeset
drm                  618496  4 drm_kms_helper,nvidia,nvidia_drm
```

Additionally, for monitoring purposes, you can install `nvidia-smi`. Simply run the following command:

```
apt -y install nvidia-smi
```

This allows you, to see your GPU's current status:

```
~# nvidia-smi
Mon Jul 11 13:33:22 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.48.07    Driver Version: 515.48.07    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Quadro P400         Off  | 00000000:03:00.0 Off |                  N/A |
| 45%   49C    P0    N/A /  N/A |      0MiB /  2048MiB |      0%      Default |
|                               |                      |                  N/A |
```

```
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

2.1 Installing CUDA Toolkit

> This only works on systems with x86_64 architecture

First, remove outdated signing keys, by running

```
apt-key del 7fa2af80
```

After that, setup the CUDA repository:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/<distro>/<arch>/cuda-keyring_1.0-1_all.deb
```

```
dpkg -i cuda-keyring_1.0-1_all.deb
```

Update your repositories and install CUDA:

```
apt update
apt -y install cuda
```

After a successful installation, reboot your system.

2.2 Compiling FFmpeg

Before you start, install the following packages, if missing:

```
apt -y install build-essential yasm cmake libtool libc6 libc6-dev unzip wget libnuma1 libnuma-dev pkg-config
```

After that, clone the ffnvcodec and FFmpeg repository.

```
git clone https://git.videolan.org/git/ffmpeg/nv-codec-headers.git
git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg/
```

Install ffnvcode:

```
cd nv-codec-headers && sudo make install && cd ../ffmpeg
```

Configure FFmpeg:

```
./configure --enable-nonfree --enable-cuda-nvcc --enable-libnpp --extra-cflags=-I/usr/local/cuda/include --extra-ldflags=-L/usr/local/cuda/lib64 --disable-static --enable-shared
```

Compile FFmpeg:

```
make -j 8 # Change according to available Threads!
```

Install libraries

```
make install
```

Check your installation, by running `ffmpeg`. If you run into the following error

```
ffmpeg: error while loading shared libraries: libavdevice.so.52: cannot open shared object file: No such file or directory
```

you need to make a change to the `/etc/ld.so.conf` config file. First, find out, where the library is located:

```
find / -name "libavdevice.so.52"
```

This returns `/usr/local/lib` as the location. Simply add this information to the above config file and run

```
ldconfig
```

You should now be able to enable Hardware Acceleration in Jellyfin. Head over to your Jellyfin Web GUI and navigate to `Administration` -> `Dashboard` -> `Playback` and set `Hardware Acceleration` to `Nvidia NVENC`. Depending on your GPU, you can enable hardware decoding for the supported Codecs. To test, if the GPU is actually doing the work, run a movie, and check `nvidia-smi` for running processes:

```
~# nvidia-smi
Mon Jul 11 13:50:25 2022
+-----------------------------------------------------------------------------+
```

```
| NVIDIA-SMI 515.48.07    Driver Version: 515.48.07    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Quadro P400         Off  | 00000000:03:00.0 Off |                  N/A |
| 37%   50C    P0    N/A /  N/A |    823MiB /  2048MiB |     66%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     61181      C   ...ib/jellyfin-ffmpeg/ffmpeg      821MiB |
+-----------------------------------------------------------------------------+
```

Revision #1
Created 24 November 2023 09:34:55 by Admin
Updated 24 November 2023 09:38:11 by Admin